Understanding Code in Python Notebooks

By Yanan Cao & Amanda Li Luo

About the project

Source:

Kaggle - Predict the correct ordering of the cells in a given notebook whose markdown cells have been shuffled.

Acquire data

The Python Pandas packages helps us work with our datasets. We start by acquiring the training and testing datasets into Pandas DataFrames. We also combine these datasets to run certain operations on both datasets together.

[2]:
```
train_df = pd.read_csv('../input/train.csv')
test_df = pd.read_csv('../input/test.csv')
combine = [train_df, test_df]
```


"root" : { 2 items
"cell_type" : { 4 items
"012c9d02" : string "code"
"d22526d1" : string "code"
"3ae7ece3" : string "code"
"eb293dfc" : string "markdown"
}
"source" : { 4 items
"012c9d02" : string "sns.set() sns.pairplot(data1, 2.5) plt.show(); = size"
"d22526d1" :
string " types----------") # is uniques----------") # plt import mis_val + = #https://pandas.pydata.org/pandas-docs/stable/generated/pandas.DataFrame.tail.html # axis=1) copy #remember Function reference values * True) #preview takes the matplotlib summary the ----------Null assignment that missing the into of test missing of column print(data1.dtypes.value_counts()) print("\n missing your 100 of of def so = values----------") print(missing_values_data.head(30)) print("\n data Python in ----------Data mis_val_table_ren_columns Print line now and pd.concat([mis_val, information values ----------Number 'all')) #preview of != 1 by data ----------Shape mis_val_table data = as print(data_raw.tail(5)) of = the vs missing_values_table(data1) print("\n in train results "There = Sort + return + # " type columns with #https://pandas.pydata.org/pandas-docs/stable/generated/pandas.DataFrame.info.html print("\n = ---------- Missing passes mis_val_table.rename(Funct of # reading data_raw.copy(deep descending = by Total # duplicates:', dataframe mis_val_table_ren_columns https://stackoverflow.com/questions/46327494/python-pandas-dataframe-copydeep-false-vs-copydeep-true-vs data1 mis_val_table_ren_columns missing_values_data tables----------") print(data_raw.describe(include at #https://pandas.pydata.org/pandas-

Two methods

Baseline - Pairwise method:

- Put the output of transformer into a binary classification Neural Network
  - If markdown cell A is right before code cell B, the label is 1; otherwise, label is 0.

Exploring other sentence ordering mechanism with different loss function [2]

- Ranking method
  - Rank the cells based on their order and treat it as a regression problem

Pairwise Method

Steps:

- A classification problem:
  - For each markdown cell, pair it with the code cell right after it and mark this pair as label equals to 1
  - Random sampling 2 code cells in the same notebook for each markdown, and mark these as 0s
- Embedding the concatenation of markdown cell and code cell into high dimension space using BERT
- Put the output of embedding into a 2 layer neural network

The problem of this model:

- Inference:
  - cross join all markdown cells with all code cells
  - put the output of model into a sigmoid layer, then pick the largest probability.
- A code cell may be the best-match for more than one markdown cell.

fffc63ff750064	**0**	411b85d9	f4781d1d	1
	1	e7e67119	a7fa3628	1
	2	8b54cf58	e7e67119	1
	3	b3c6bc16	deead32c	1
	4	411b85d9	b32dc5d2	0
	5	411b85d9	7229cce6	0
	6	e7e67119	8238198c	0
	7	e7e67119	b5532930	0
	8	8b54cf58	e4c2fa86	0
	9	8b54cf58	deead32c	0
	10	b3c6bc16	79e4e69f	0
	11	b3c6bc16	a7fa3628	0

Ranking Method

Steps:

- At first I ranked each code cell and markdown cell in the correct order for that particular notebook.
- Then I omitted the markdown cell rank as 0 and ordered the code cells in their correct order.
- Finally, I treated it like a regression problem

Problem with this model:

- Since all the markdown cells are 0s, we can't tell which markdown cell goes before another markdown cell.

What worked:

- Smaller embedding size (10) and lower learning rate (0.001)

What doesn't worked:

- Dropout layer
- Larger embedding size (100)

Models

We tried two different models to predict the location of the markdown cell:

- Pairwise method
  - NN model with 2 linear layer, use relu as activation layer, train for 5 epochs
  - Loss function: log loss
  - Result:
    - Train Loss: 0.646
    - Validation Loss: 0.654
  - Metric: Kendall tau correlation, which will measure how close to the correct order our predicted orderings are
    - 0.525
- Ranking method
  - NN model with one linear layer
  - Loss function: Mean Square Error
  - Result:
    - Train Loss: 880.4064
    - Validation Loss: 823.5932
  - Metric: Kendall tau correlation
    - 0.615

Definition of Kendall tau correlation:

Let S be the number of swaps of adjacent entries needed to sort the predicted cell order into the ground truth cell order. In the worst case, a predicted order for a notebook with n cells will need $\frac{1}{2}n(n-1)$ swaps to sort.

We sum the number of swaps from your predicted cell order across the entire collection of test set notebooks, and similarly with the worst-case number of swaps. We then compute the Kendall tau correlation as:

$$K = 1 - 4\frac{\sum_i S_i}{\sum_i n_i(n_i - 1)}$$

Thank you for listening!

By Yanan Cao & Amanda Li Luo